

Electron Lens Models

Ivan Morozov*

Fermi National Accelerator Laboratory
Batavia, IL, 60510, USA

*imorozov@fnal.gov

Contents

1 DRIFT-KICK INTEGRATOR	4
2 HOLLOW ELECTRON-BEAM LENS MODELS	4
2.1 AZIMUTHAL MODEL	4
2.2 RADIAL MODEL	8
3 SIMULATION OF AZIMUTHAL MODEL	9
3.1 INDIVIDUAL HARMONICS EFFECTS	9
3.2 LOST PARTICLES DISTRIBUTION	11
3.3 ANTIPIRON BEAM DENSITY	12
3.4 ALTERING LATTICE TUNES	13
3.5 FMA	14
4 ROUTINE DESCRIPTION	17
4.1 MULTIPOLE CORE MODEL	17
4.2 IDEAL LENS	18
4.3 RADIAL LENS	19

List of Figures

1	Transverse electron beam profile from measurments.	5
2	Multipole fields for $m = 1, 2, 3$.	6
3	Angular distribution function and it's FFT.	7
4	Radial distribution function.	8
5	Profile interpolation.	8
6	Beam normalized intensity for $skip = 1/1, 1/2, 1/5$.	10
7	Invariant amplitudes without lens.	11
8	Invariant amplitudes with lens ($m = 2, skip = 1/5$).	11
9	Beam density evolution. with lens – (left); without lens – (right)	12
10	Horizontal emmitance.	13
11	Verticle emmitance.	13
12	Beam intensity for different lattice tunes.	14
13	FMA in amplitude space with lens	15
14	FMA in amplitude space without lens	15
15	FMA in frequency space with lens	16
16	FMA in frequency space without lens	16

1 DRIFT-KICK INTEGRATOR

We use a simple symplectic drift-kick integrator for electron lens modeling.

$$p_{n+1} = p_n + f(x_n), \quad x_{n+1} = x_n + f(p_{n+1})$$

Electron lens is treated as a conventional transverse element. Transverse Lorentz force acting on a test particle is

$$F_x = qE_x - q\beta_{\bar{p}}cB_y, \quad F_y = qE_y + q\beta_{\bar{p}}cB_x$$

Then we express this force in terms of electron beam rest frame electric fields,

$$F_x = q\gamma_b(1 + \beta_b\beta_{\bar{p}})E_x^R, \quad F_y = q\gamma_b(1 + \beta_b\beta_{\bar{p}})E_y^R$$

where γ_b , β_b are electron beam relativistic parameters, $\beta_{\bar{p}}$ is antiproton relative speed. Then transverse kicks are the following,

$$\Delta x' = \Delta s \frac{F_x}{\beta_{\bar{p}}cp_{\bar{p}}}, \quad \Delta y' = \Delta s \frac{F_y}{\beta_{\bar{p}}cp_{\bar{p}}}$$

2 HOLLOW ELECTRON-BEAM LENS MODELS

We use two models to include transverse imperfections. The first one is due to the lack of azimuthal symmetry of the beam. The second one is due to non-ideal beam radial profile.

2.1 AZIMUTHAL MODEL

In this case we want to investigate how angular imperfections influence the antiproton beam core. Electron lens is assumed to be infinitely long and thin with charge distribution,

$$\rho(r, \theta) = \frac{f(\theta)}{2\pi r_b} \delta(r - r_b)$$

where r_b is cylinder (beam) radius, $f(\theta)$ is some function that gives angular distribution shape. Then we express charge distribution in the following form,

$$\rho(r, \theta) = \frac{I_b}{\beta_b c} \frac{\delta(r - r_b)}{2\pi r_b} \sum_{m=0}^{\infty} \xi_m \cos(m\theta + \phi_m)$$

where ξ_m is a relative amplitude of m -th harmonic (relative to zero harmonic's amplitude, i.e. to the beam line charge density), ϕ_m is harmonic's phase, I_b is electron beam current, β_b is electron velocity. These parameters are to be determined from real transverse beam profile (Fig. 1). For this distribution electric field can be obtained for each harmonic.

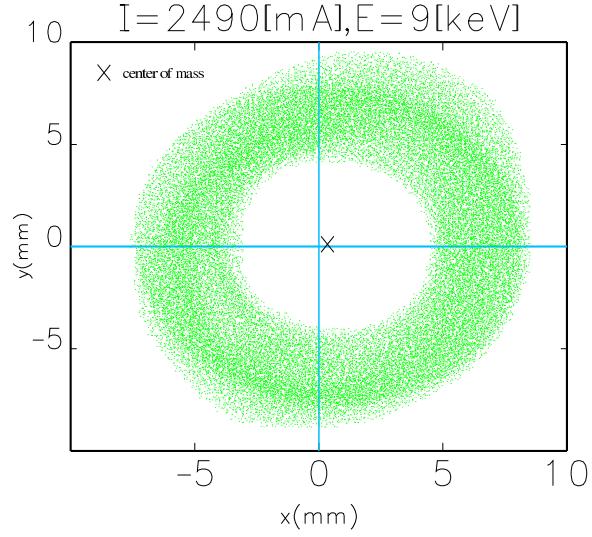
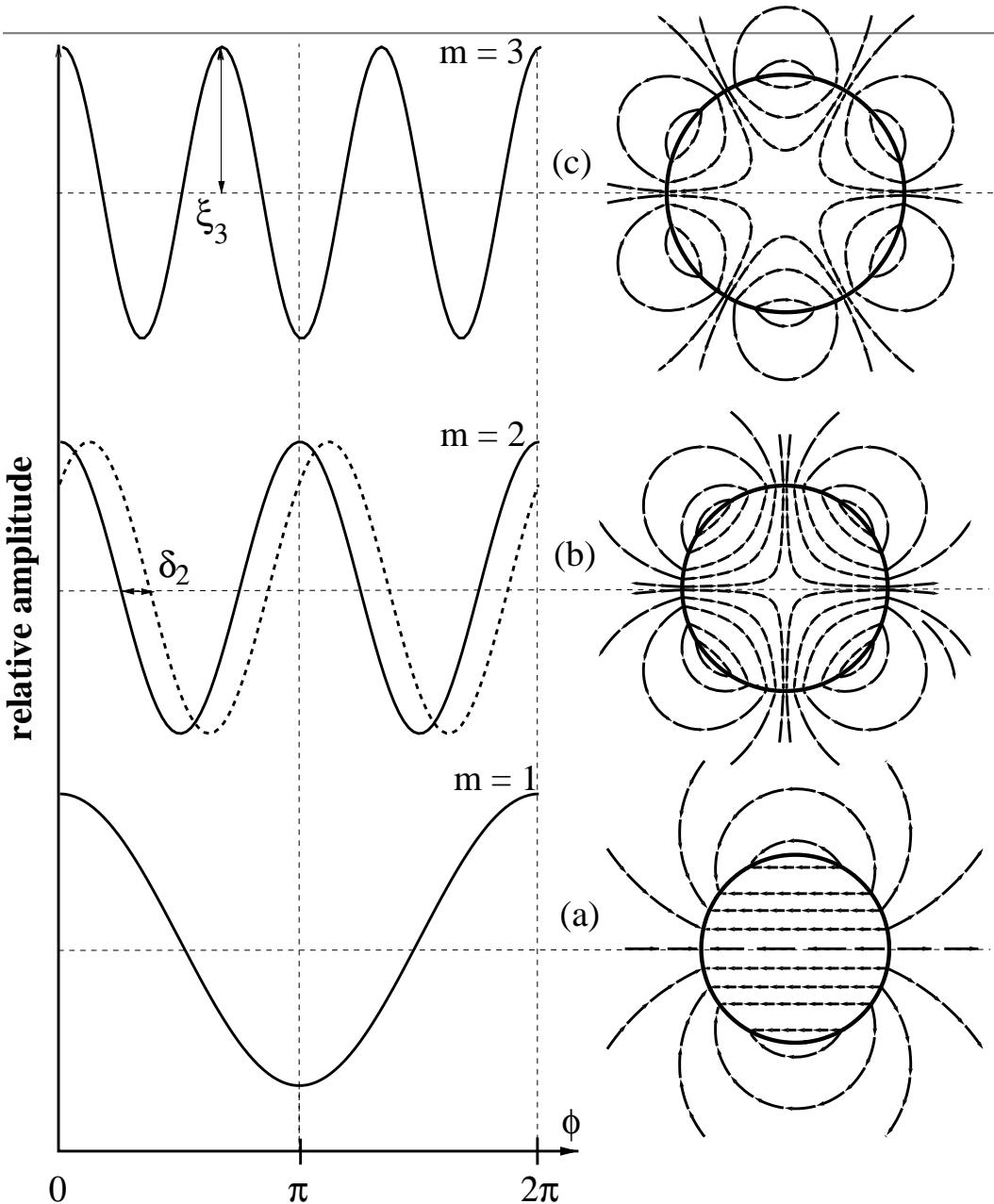


Figure 1: Transverse electron beam profile from measurements.

$$E_r(r, \theta) = \frac{I_b \xi_m}{4\pi \varepsilon_0 \beta_b} \cos(m\theta + \phi_m) \begin{cases} -\frac{r^{m-1}}{r_b^m}, & r < r_b \\ r^{m-1} r_b^m, & r > r_b \end{cases} \quad (1a)$$

$$E_\theta(r, \theta) = \frac{I_b \xi_m}{4\pi \varepsilon_0 \beta_b} \sin(m\theta + \phi_m) \begin{cases} \frac{r^{m-1}}{r_b^m}, & r < r_b \\ r^{m-1} r_b^m, & r > r_b \end{cases} \quad (1b)$$

On Fig. 2 electric field is shown for $m = 1, 2, 3$. To determine harmonic parameters we put origin into center-of-charge and divide space into angular slices. Then we count the number of particles in each slice and normalize this value and perform FFT to determine amplitudes and phases (Fig. 3). Transverse kicks for this model are,

Figure 2: Multipole fields for $m = 1, 2, 3$.

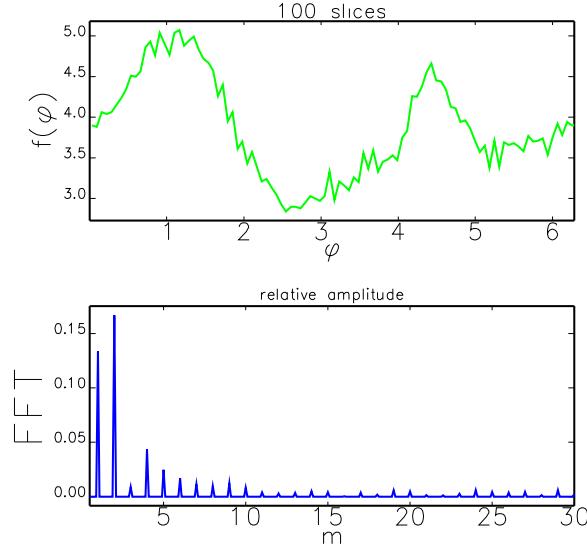


Figure 3: Angular distribution function and it's FFT.

$$\Delta x' = -\Omega \Delta s \xi_m \frac{r^{m-1}}{r_b^m} \cos((m-1)\theta + \delta_m) \quad (2)$$

$$\Delta y' = \Omega \Delta s \xi_m \frac{r^{m-1}}{r_b^m} \sin((m-1)\theta + \delta_m) \quad (3)$$

where,

- $\Omega = 0.3 \times 10^{-7} \frac{I(A)}{p_{\bar{p}}(GeV/c)} \gamma_b \frac{1+\beta_b \beta_{\bar{p}}}{\beta_b \beta_{\bar{p}}}$
- Δs – kick-drift section length
- I_b – electron beam current
- $p_{\bar{p}}$ – antiproton longitudinal momenta
- $\beta_b, \beta_{\bar{p}}$ – electrons and antiprotons relative velocities
- ξ_m – harmonic's relative amplitude
- δ_m – harmonics phase
- r_b – electron beam radius

2.2 RADIAL MODEL

As far radial electron beam distribution can is not ideal (Fig. 4) we should include it into electron lens model. This can be done by using interpolation of radial

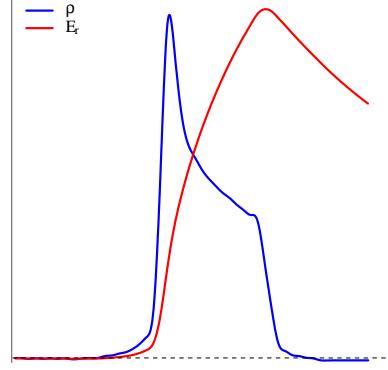


Figure 4: Radial distribution function.

profile. Profile is divided into two regions in order to have a better interpolation for each region (Fig.5) and to combine with azimuthal model. Then transverse kick is,

$$\Delta r' = 2\Omega r_m f(r) \Delta s$$

where Ω is the same parameter as for azimuthal model, $f(r)$ is normalized electric field in a.u., r_m is radius where $f(r)$ value is maximum.

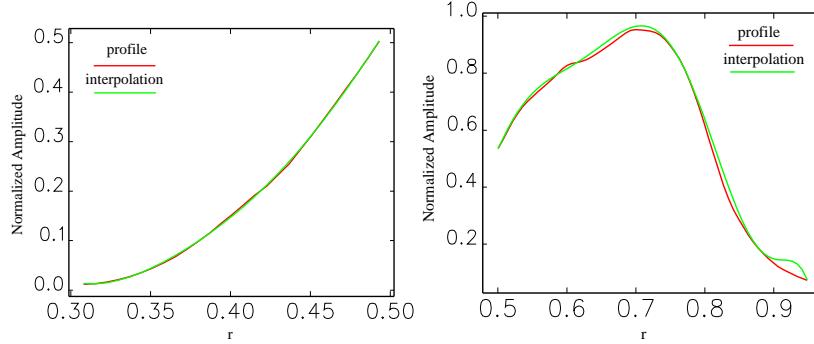


Figure 5: Profile interpolation.

3 SIMULATION OF AZIMUTHAL MODEL

For azimuthal model several simulations were performed to investigate effects on the antiproton beam core:

- individual harmonics contributions
- current are tune scan for second harmonic with $skip = 1/5$
- FMA analysis for second harmonic with $skip = 1/5$

3.1 INDIVIDUAL HARMONICS EFFECTS

Runs were performed for the case when electron lens is presented as a single drift-kick section for harmonics $m = 1, 2, 3, \dots, 20$ with lattice tunes $Q_x = 0.578$, $Q_y = 0.575$ and the following lens parameters,

- $L_b = 200(\text{cm})$ – lens length
- $slice = 1$ – number of drift-kick sections
- $\beta_b = 0.2$ – relative electron velocity (energy about 10keV)
- $I = 0.05, 0.3, 0.5, 1.0, \dots, 3.0(?)$ – beam current
- $skip = 1/1, 1/2, 1/3, \dots, 1/8$ – pulse pattern
- $m = 1, 2, \dots, 20$ – harmonics
- 3×10^6 – number of tunes (about 1 minute of real time)

The only significant effect was observed for second harmonic (the quadrupole one) with $skip = 1/5$, i.e. lens is "on" every 6th pass of antiproton beam. On Fig. 6 the normalized beam intensity is shown for the case when lens was treated as a sum of first 5, 10, 20 harmonics. This plot shows that effect appears only with $skip = 1/5$ and in fact only for quadrupole harmonic.

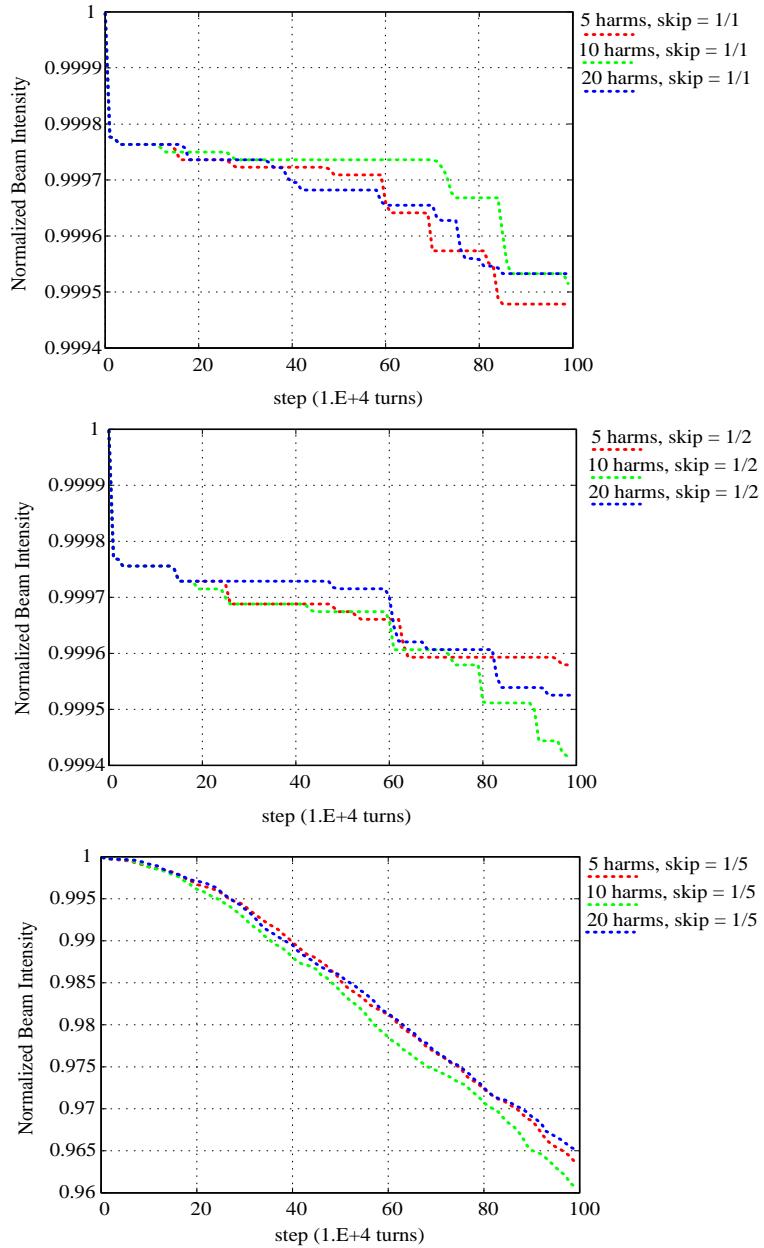


Figure 6: Beam normalized intensity for $\text{skip} = 1/1, 1/2, 1/5$.

3.2 LOST PARTICLES DISTRIBUTION

In order to identify losses we plot particles invariant amplitudes ($X = \sqrt{x^2 + x'^2}$) in initial distribution. One can see that losses appear in the range of $2.5 - 4\sigma$.

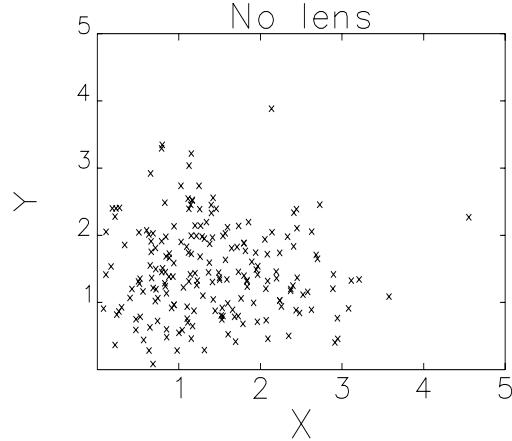


Figure 7: Invariant amplitudes without lens.

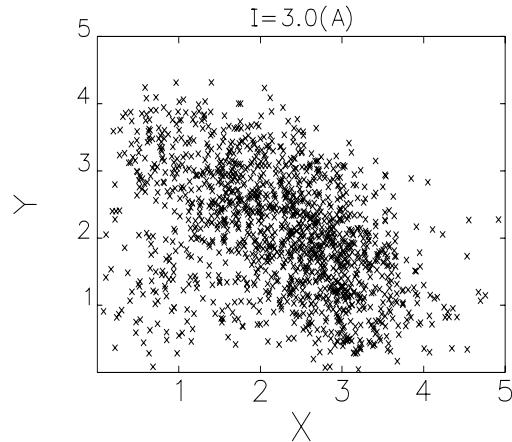


Figure 8: Invariant amplitudes with lens ($m = 2$, $skip = 1/5$).

3.3 ANTIPROTON BEAM DENSITY

Another way to identify losses is to watch how beam density changes when electron lens is introduced. On Fig. 9 beam density is shown for several simulation steps (10^4 turns is each step). Each line correspond shows the points with equal density. Density decays as \sqrt{e} from line to line. It is clear that lost particles come from low density regions with high amplitudes.

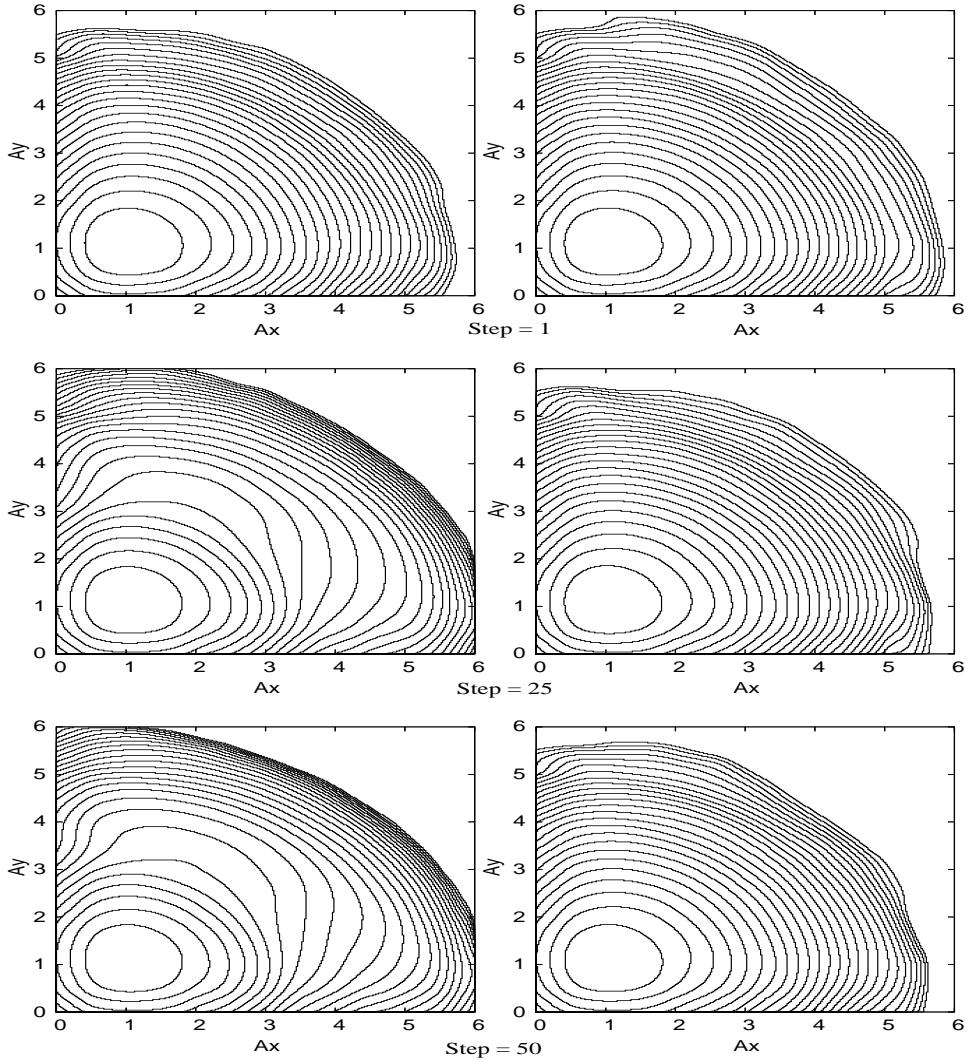


Figure 9: Beam density evolution. with lens – (left); without lens – (right)

3.4 ALTERING LATTICE TUNES

In order to find the reason of particle loss we check how lattice tunes influence the simulations. Figs. 10, 11 and 12 show that lens effect from second harmonic with $skip = 1/5$ disappears and no emittance growth is introduced.

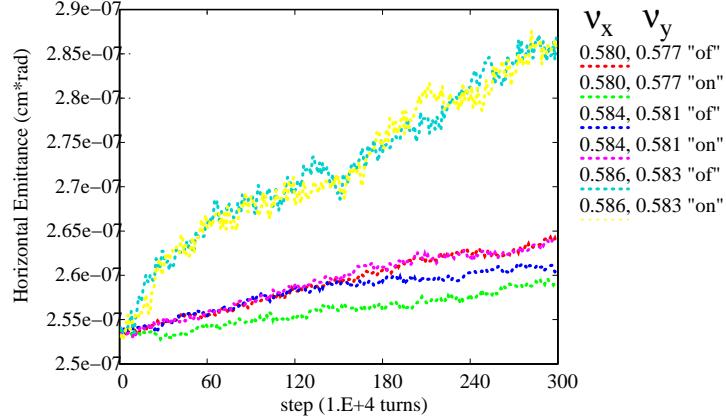


Figure 10: Horizontal emmittance.

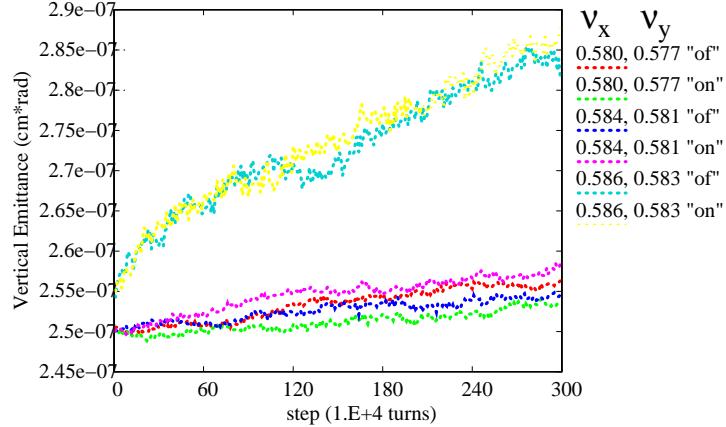


Figure 11: Verticle emmittance.

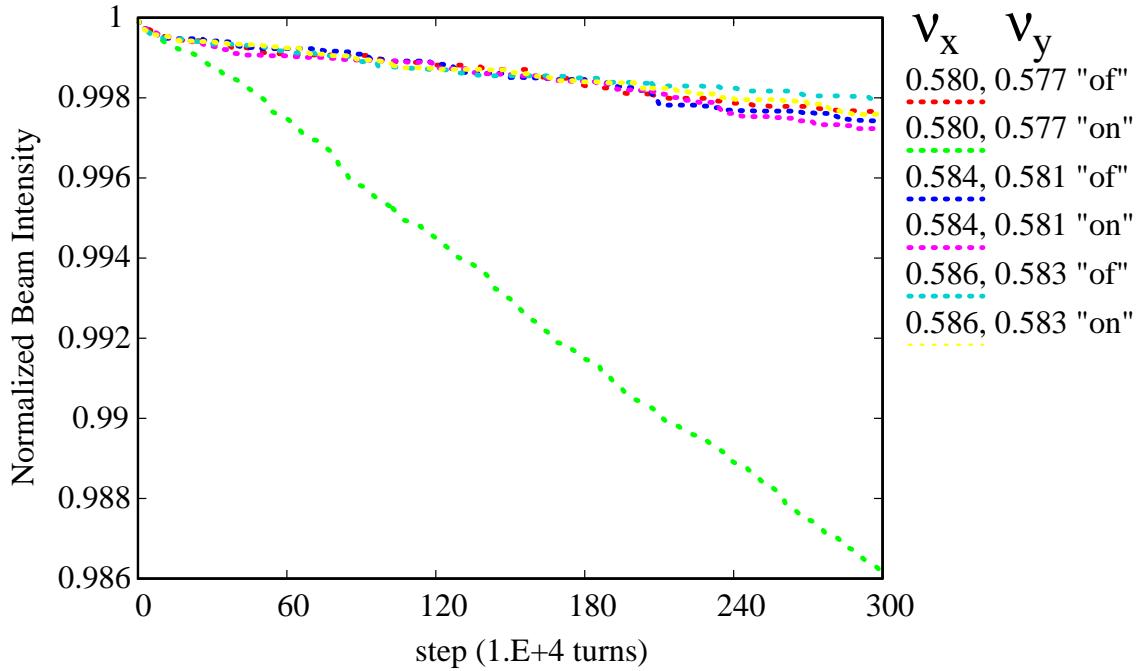


Figure 12: Beam intensity for different lattice tunes.

3.5 FMA

FMA was performed to searcher for resonances that may be responsible for particles losses at big amplitudes. On Fig. 13 and 14 FMA is shown in amplitude space and in frequency space on 15 and 16. In the case when lens in on with only quadrupole mode $m = 2$ and $skip = 1/5$ 12th order resonances appear. In amplitude space this resonances are seen in the region were we have losses.

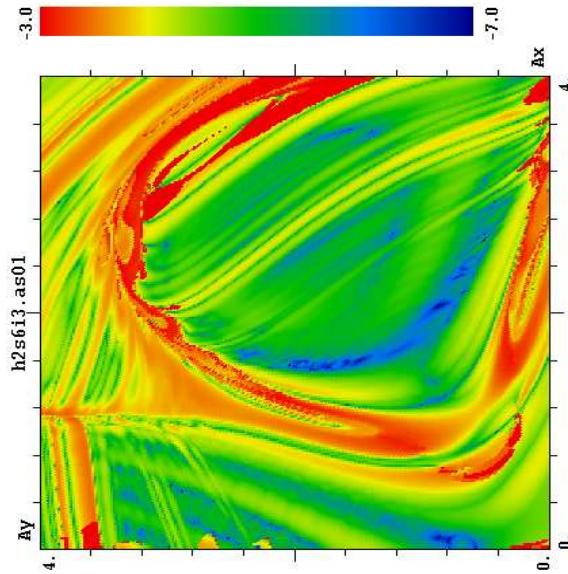


Figure 13: FMA in amplitude space with lens

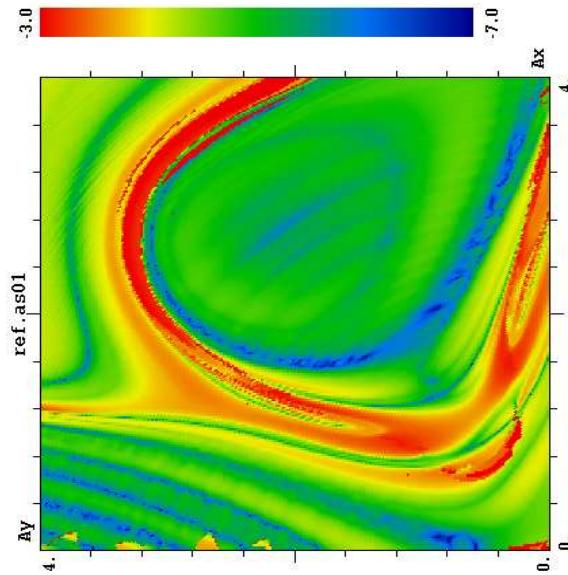


Figure 14: FMA in amplitude space without lens

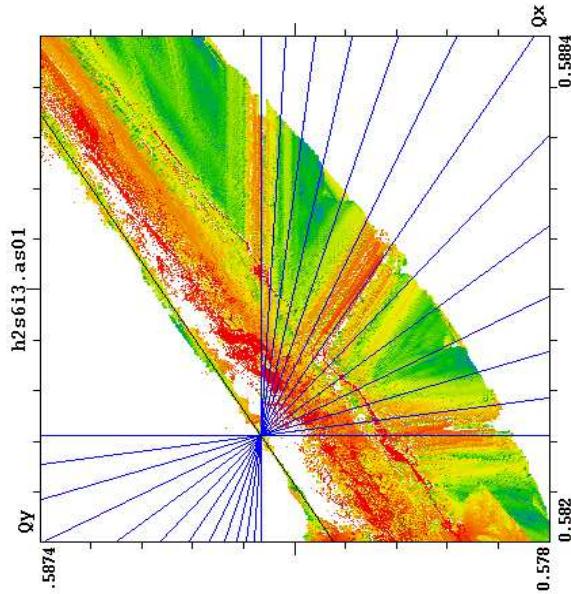


Figure 15: FMA in frequency space with lens

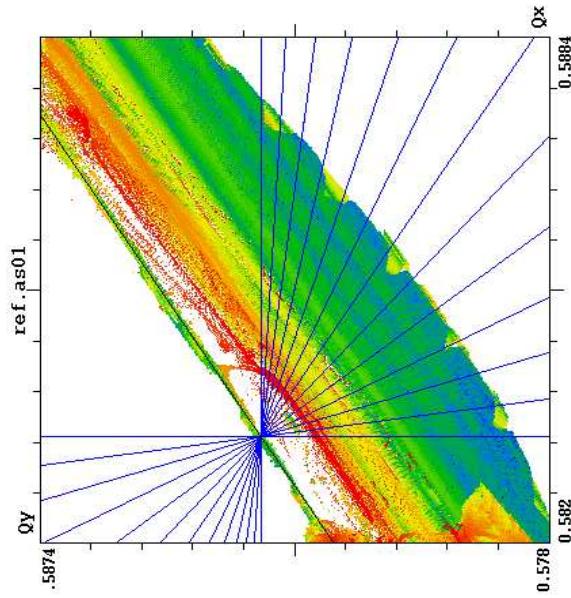


Figure 16: FMA in frequency space without lens

4 ROUTINE DESCRIPTION

Currently models of electron lens are built in Lifetrack as external elements. All external elements are stored in "trac_ext.f90" file. Lifetrack allows to use 20 parameters (array "params") for description of external element with are used in the input lattice file. Lifetrack passes six coordinates ($x[cm], x', y[cm], y', z[cm], \delta$) in array "coord".

Input coordinates comes to the center of electron lens element. That's why we should do a "back-tracking" to the entrance of lens first and after tracking we "back-track" to the center again.

Another feature due to Liftrac using is a condition for nturns=0. nturns is an integer that is used in Lifetrack to count the number of turns. The case nturns=0 corresponds to program initializing when Liftrack performs close orbit calculations and other. The lens is off at this step because Liftrack uses a particle with zero coordinates for orbit calculations but this cases an error in lens since it includes calculations of trigonometric functions.

"track_ext" contains two subroutines where electron lens description is needed. First subroutine PRE_TRACK_EXT performs precalculations with user specified parameters (Values in input file). Second subroutine TRACK_EXT performs tracking.

4.1 MULTIPOLE CORE MODEL

This subroutine is used for core study. It performs transverse kicks in a region $r < r_b$.

Input file:

Value_1: param(1) - skip
Value_2: param(2) - # of slices
Value_3: param(3) - length[cm]
Value_4: param(4) - Ω
Value_5: param(5) - r_b [cm]
Value_6: param(6) - harmonic number
Value_7: param(7) - harmonic amplitude
Value_8: param(8) - haronic phase

PRE_TRACK_EXT:

CASE('EXT_LEN1')

param(20)=param(3)/param(2) ! slice length [cm]

param(3)=-param(3)/2..8 ! minus half lens length for back tracking

```

param(4)=param(4)*param(20)*param(7)/param(5)**param(6) ! kick factor
param(6)=(param(6)-1._8)/2._8 ! modified harm number
TRACK_EXT:
CASE('EXT_LEN1') ! ONLY CORE MODEL
IF(MOD(nturn,INT(param(1)))==0 .AND. nturn/=0) THEN ! start condition
! back tracking
coord(1)=coord(1)+param(3)*coord(2)
coord(3)=coord(3)+param(3)*coord(4)
! lens tracking
DO i=1,param(2),1
r=param(4)*(coord(1)*coord(1)+coord(3)*coord(3))**param(6)
t=2._8*param(6)*ATAN2(coord(3),coord(1))+param(8)
coord(2)=coord(2)-r*COS(t)
coord(4)=coord(4)+r*SIN(t)
coord(1)=coord(1)+param(20)*coord(2)
coord(3)=coord(3)+param(20)*coord(4)
END DO
! back tracking
coord(1)=coord(1)+param(3)*coord(2)
coord(3)=coord(3)+param(3)*coord(4)
END IF

```

4.2 IDEAL LENS

This is an ideal lens element. Value_1: param(1) - skip

Value_2: param(2) - # of slices

Value_3: param(3) - length[cm]

Value_4: param(4) - Ω

Value_5: param(5) - r_{inner} [cm]

Value_6: param(6) - r_{outer} [cm]

PRE_TRACK_EXT:

CASE('EXT_LEN5')

param(20)=param(3)/param(2) ! slice length [cm]

param(3)=-param(3)/2._8 ! minus half lens length for back tracking

param(4)=param(4)*param(20)*2.0

TRACK_EXT:

CASE('EXT_LEN5') ! IDEAL LENS

```

! START CONDITION
IF(MOD(nturn,INT(param(1)))==0 .AND. nturn/=0) THEN
! BACK TRACKING
coord(1)=coord(1)+param(3)*coord(2)
coord(3)=coord(3)+param(3)*coord(4)
! LENS TRACKING
DO i=1,param(2),1
r=SQRT(coord(1)*coord(1)+coord(3)*coord(3))
t=ATAN2(coord(3),coord(1))
IF (r .LT. param(5)) THEN
r=0.0
ELSEIF (r .GT. param(6)) THEN
r=param(4)/r
ELSE
r=param(4)*(r**2-param(5)**2)/(r*(param(6)**2-param(5)**2))
ENDIF
coord(2)=coord(2)+r*COS(t)
coord(4)=coord(4)+r*SIN(t)
coord(1)=coord(1)+param(20)*coord(2)
coord(3)=coord(3)+param(20)*coord(4)
END DO
! BACK TRACKING
coord(1)=coord(1)+param(3)*coord(2)
coord(3)=coord(3)+param(3)*coord(4)
END IF

```

4.3 RADIAL LENS

This is an element that uses interpolated radial profile. Interpolation polynomials are build in. Value_1: param(1) - skip
Value_2: param(2) - # of slices
Value_3: param(3) - length[cm]
Value_4: param(4) - Ω
Value_5: param(5) - r[cm] boundary between polynomials
Value_6: param(13) - r[cm] beginning of first poly
PRE_TRACK_EXT:
CASE('EXT_LEN8') ! INSIDE/OUTSIDE

```

param(20)=param(3)/param(2)
param(3)=-param(3)/2._8
param(10)=param(4)*2.0*param(20)/param(15)
TRACK_EXT:
CASE('EXT_LEN8')
IF(MOD(nturn,INT(param(1)))==0 .AND. nturn/=0) THEN
! TRACK PARTICLE BACK
coord(1)=coord(1)+param(3)*coord(2)
coord(3)=coord(3)+param(3)*coord(4)
DO i=1,param(2),1
r=SQRT(coord(1)**2+coord(3)**2)
t=ATAN2(coord(3),coord(1))
IF(r .LT. param(5))THEN
IF(r .GT. param(13)) THEN
p=param(10)*(-1.76026 +78.6339*r-1515.51*r**2 +16539.3*r**3-&
112683*r**4+497416*r**5-1.42429e+06*r**6 +2.55448e+06*r**7-&
2.60782e+06*r**8+1.15636e+06*r**9)
ELSE
p=0.0
ENDIF
ELSE
p=param(10)*(-230.976 +2108.63*r-8098.24*r**2 +16916.5*r**3-&
20675.1*r**4 +14748*r**5-5674.02*r**6 +906.087*r**7)
ENDIF
coord(2)=coord(2)+p*COS(t)
coord(4)=coord(4)+p*SIN(t)
coord(1)=coord(1)+param(20)*coord(2)
coord(3)=coord(3)+param(20)*coord(4)
END DO
! TRACK PARTICLE BACK
coord(1)=coord(1)+param(3)*coord(2)
coord(3)=coord(3)+param(3)*coord(4)
END IF

```